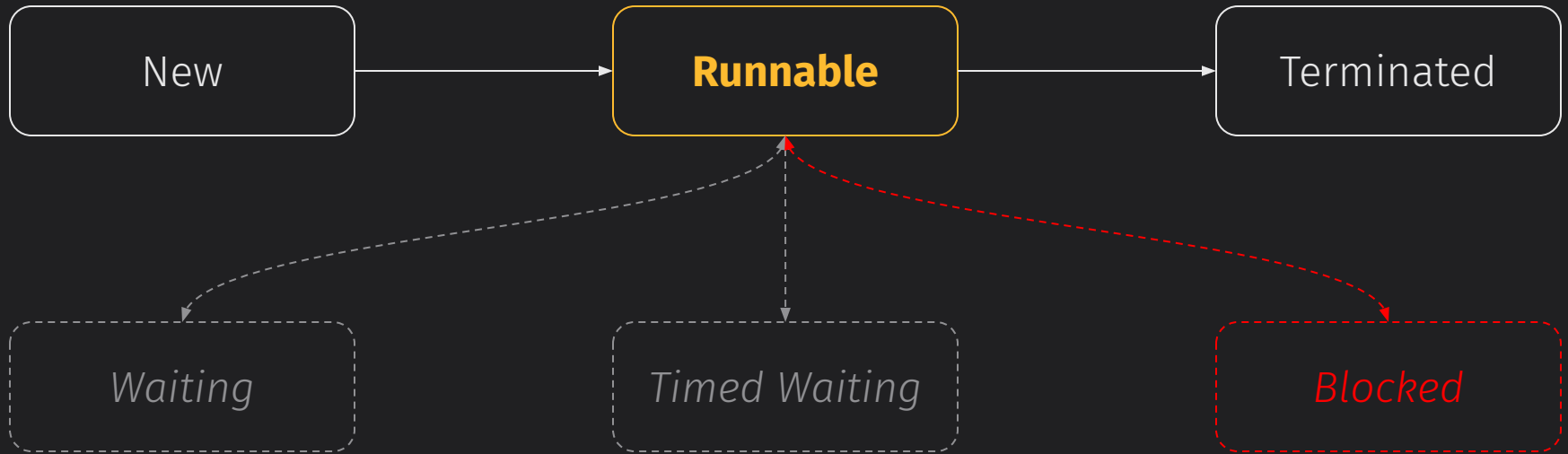


# Thread Lifecycle

## CS 272 Software Development

# Thread States



<https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/lang/Thread.State.html>



## CODE (SIMPLIFIED)

```
public ThreadStateDemo() {
    parent = Thread.currentThread();
    worker = new Worker();

    worker.start();
    calculate(500);

    worker.join();
}

private class Worker extends Thread {
    public void run() {
        calculate(1000);
    }
}
```

## PARENT THREAD

## WORKER THREAD



## CODE (SIMPLIFIED)

```
public ThreadStateDemo() {  
    parent = Thread.currentThread();  
    worker = new Worker();  
  
    worker.start();  
    calculate(500);  
  
    worker.join();  
}  
  
private class Worker extends Thread {  
    public void run() {  
        calculate(1000);  
    }  
}
```

## PARENT THREAD

RUNNABLE

|

## WORKER THREAD



## CODE (SIMPLIFIED)

```
public ThreadStateDemo() {
    parent = Thread.currentThread();
    worker = new Worker();

    worker.start();
    calculate(500);

    worker.join();
}

private class Worker extends Thread {
    public void run() {
        calculate(1000);
    }
}
```

## PARENT THREAD

RUNNABLE  
|  
new Worker() ————— NEW  
|  
:

## WORKER THREAD



## CODE (SIMPLIFIED)

```
public ThreadStateDemo() {
    parent = Thread.currentThread();
    worker = new Worker();

    worker.start();
    calculate(500);

    worker.join();
}

private class Worker extends Thread {
    public void run() {
        calculate(1000);
    }
}
```

## PARENT THREAD

RUNNABLE  
|  
new Worker() —————  
|  
worker.start()  
|

## WORKER THREAD

NEW  
⋮  
RUNNABLE  
|



## CODE (SIMPLIFIED)

```
public ThreadStateDemo() {
    parent = Thread.currentThread();
    worker = new Worker();

    worker.start();
    calculate(500);

    worker.join();
}

private class Worker extends Thread {
    public void run() {
        calculate(1000);
    }
}
```

## PARENT THREAD

```
RUNNABLE
|
new Worker()
|
worker.start()
|
calc(500)
|
```

## WORKER THREAD

```
NEW
⋮
RUNNABLE
|
calc(1000)
|
```



## CODE (SIMPLIFIED)

```
public ThreadStateDemo() {
    parent = Thread.currentThread();
    worker = new Worker();

    worker.start();
    calculate(500);

    worker.join();
}

private class Worker extends Thread {
    public void run() {
        calculate(1000);
    }
}
```

## PARENT THREAD

```
RUNNABLE
|
new Worker()
|
worker.start()
|
calc(500)
|
worker.join()
|
WAITING
|
⋮
```

## WORKER THREAD

```
NEW
⋮
RUNNABLE
|
calc(1000)
|
⋮
```





## CODE (SIMPLIFIED)

```
public ThreadStateDemo() {  
    parent = Thread.currentThread();  
    worker = new Worker();  
  
    worker.start();  
    calculate(500);  
  
    worker.join();  
}  
  
private class Worker extends Thread {  
    public void run() {  
        calculate(1000);  
    }  
}
```

## PARENT THREAD

RUNNABLE  
|  
new Worker() —————  
|  
worker.start()  
|  
calc(500)  
|  
worker.join()  
|  
WAITING  
⋮  
RUNNABLE —————  
|  
TERMINATED

## WORKER THREAD

NEW  
⋮  
RUNNABLE  
|  
calc(1000)  
|  
TERMINATED



## CODE (SIMPLIFIED)

```
public ThreadStateDemo() {
    parent = Thread.currentThread();
    worker = new Worker();

    worker.start();
    calculate(500);

    worker.join();
}

private class Worker extends Thread {
    public void run() {
        calculate(1000);
    }
}
```

## PARENT THREAD

RUNNABLE  
|  
new Worker()  
|  
worker.start()  
|  
calc(500)  
|  
worker.join()  
|  
WAITING  
|  
RUNNABLE  
|  
TERMINATED

## WORKER THREAD

NEW  
|  
RUNNABLE  
|  
calc(1000)  
|  
TERMINATED





---

CHANGE THE WORLD FROM HERE